

아파트 실거래가 예측

Date : 2020-09-10

DNA 1조

심우열 교수영 유소영

CONTENTS



1. 개요



2. Data EDA



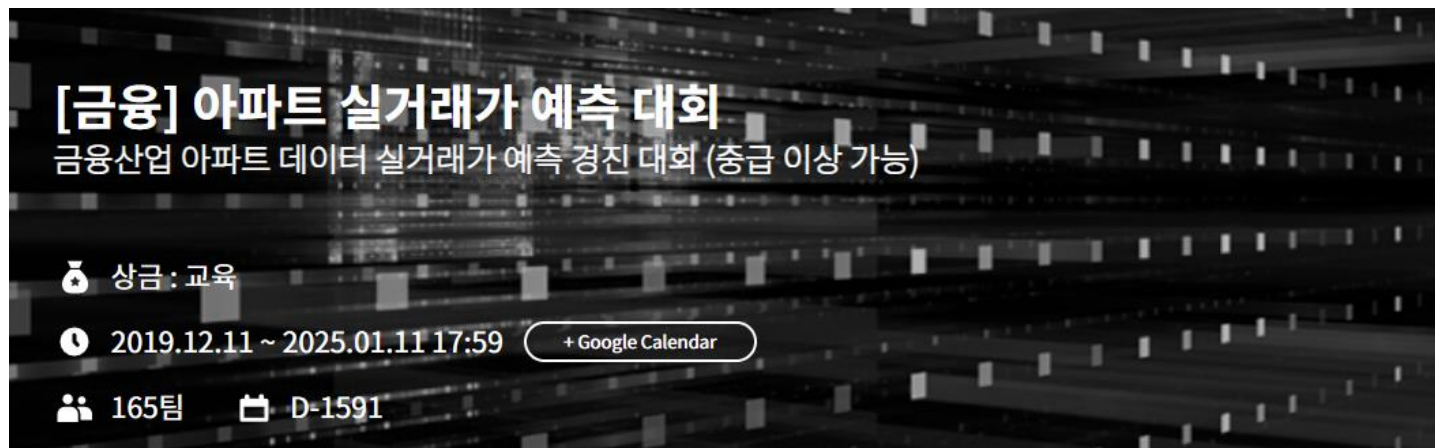
3. Preprocessing & Cleansing



4. Modeling & Conclusion

개요

- 데이콘에서 열린 아파트 실거래가 예측 대회가 교육용으로 출시되어 제공되는 데이터를 이용해 통계적 기반을 바탕으로 변수를 선택하고 머신러닝 모델들을 직접 사용해보는 시간을 가지려 함.



변수설명

➤ 변수들을 확인하고 방향성을 제시

✓ train.csv

Column	Describe
Transaction_id	아파트 거래에 대한 유니크한 아이디
Apartment_id	아파트 아이디
City	도시 (서울/부산)
Dong	동
Jibun	지번
Apt	아파트 단지 이름
Addr_kr	주소
Exclusive_use_area	전용면적
Year_of_completion	건립일자
Transaction_year_month	거래년월
Transaction_date	거래날짜
Floor	층
transaction_real_price	실거래가

✓ test.csv

Column	Describe
Transaction_id	아파트 거래에 대한 유니크한 아이디
Apartment_id	아파트 아이디
City	도시 (서울/부산)
Dong	동
Jibun	지번
Apt	아파트 단지 이름
Addr_kr	주소
Exclusive_use_area	전용면적
Year_of_completion	건립일자
Transaction_year_month	거래년월
Transaction_date	거래날짜
Floor	층

변수설명

➤ 데이터타입과 결측값을 확인

✓ Data type

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1216553 entries, 0 to 1216552
Data columns (total 13 columns):
transaction_id      1216553 non-null int64
apartment_id        1216553 non-null int64
city                1216553 non-null object
dong                1216553 non-null object
jibun               1216553 non-null object
apt                 1216553 non-null object
addr_kr             1216553 non-null object
exclusive_use_area  1216553 non-null float64
year_of_completion  1216553 non-null int64
transaction_year_month 1216553 non-null int64
transaction_date     1216553 non-null object
floor               1216553 non-null int64
transaction_real_price 1216553 non-null int64
dtypes: float64(1), int64(6), object(6)
memory usage: 120.7+ MB
```

✓ 결측값

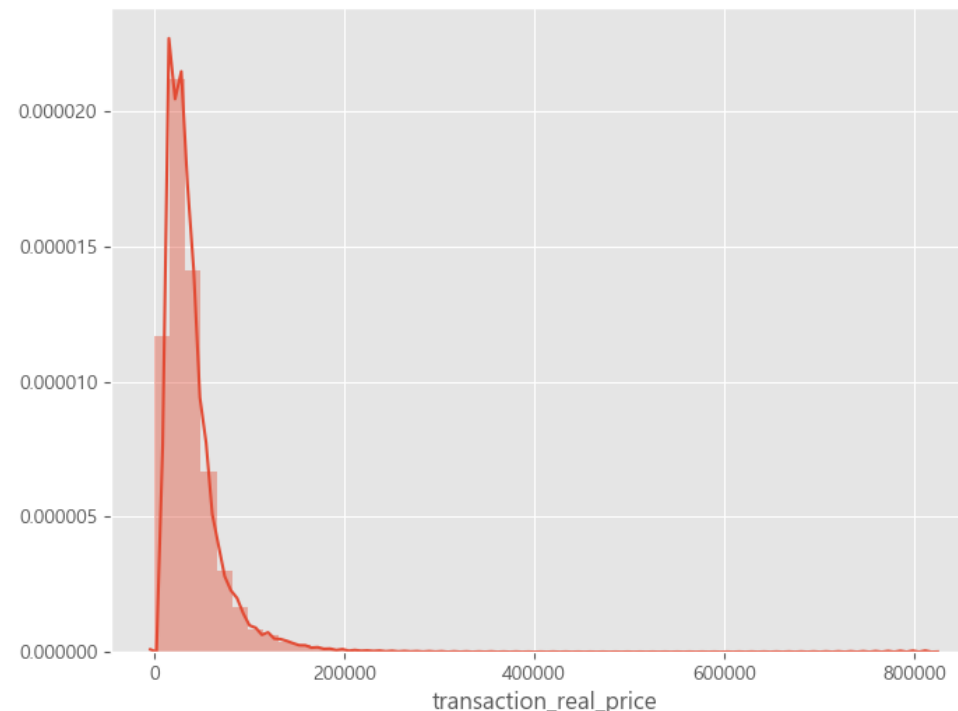
```
In [18]: train.isna().sum()
Out[18]:
transaction_id      0
apartment_id        0
city                0
dong                0
jibun               0
apt                 0
addr_kr             0
exclusive_use_area  0
year_of_completion  0
transaction_year_month 0
transaction_date     0
floor               0
transaction_real_price 0
dtype: int64
```

Dong/city/transaction_date 변수의 Labeling 과정 필요

EDA(Exploratory Data Analysis)

➤ Target의 분포를 확인하고 성능향상을 위해 변형시킴

```
count    1.216553e+06  
mean     3.822769e+04  
std      3.104898e+04  
min      1.000000e+02  
25%     1.900000e+04  
50%     3.090000e+04  
75%     4.700000e+04  
max      8.200000e+05  
Name: transaction_real_price, dtype: float64
```

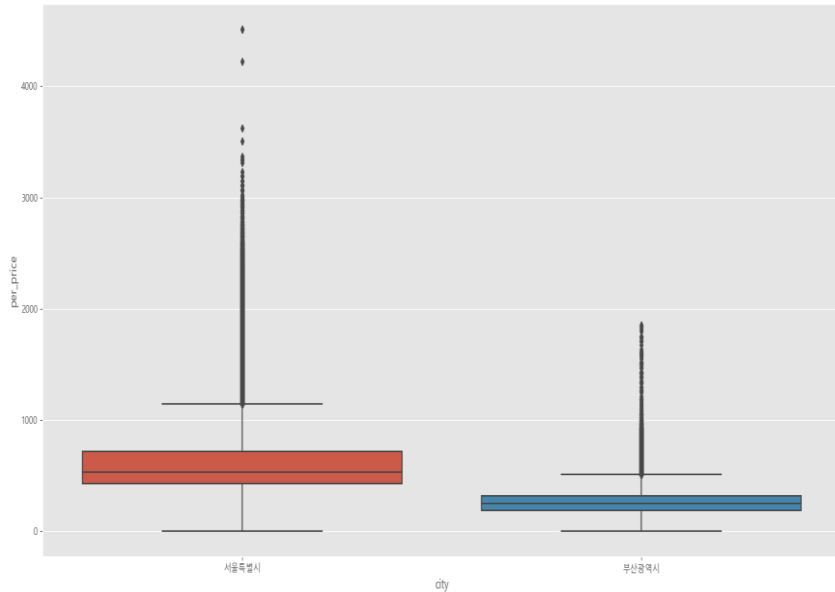


- ✓ 값의 편차가 매우 크고 왼쪽으로 몰려있는(Skewness: 3.407169) 분포를 형성
- ✓ Skewed 되어있는 값을 그대로 학습시키면 꼬리 부분이 상대적으로 모델에 영향이 거의 없이 학습됨
- ✓ 이를 방지하기 위해 target인 '실거래액'을 log화 해줌 `[np.log1p(transaction_price)]`
- ✓ 집단 비교는 (실거래액 / 면적) -> 면적당 거래액으로 비교함

EDA(Exploratory Data Analysis)

➤ 변수 'city'

✓ 부산(474,268) vs 서울 (742,285)



	df	Sum_sq	Mean_sq	F	Pr(>F)
City	1	3.8e+10	3.8e+10	671017.13	0.0
Residual	121655 1	6.8e+10	5.6e+04	NaN	NaN

- ✓ 박스플롯을 보았을 때, 서울특별시와 부산광역시는 차이가 있어 보임
- ✓ 이를 검정하기 위해 독립표본-t검정을 실시
- ✓ P-value = 0.0으로 귀무가설을 기각, 두 집단에는 평균 차이가 있다고 할 수 있음

EDA(Exploratory Data Analysis)

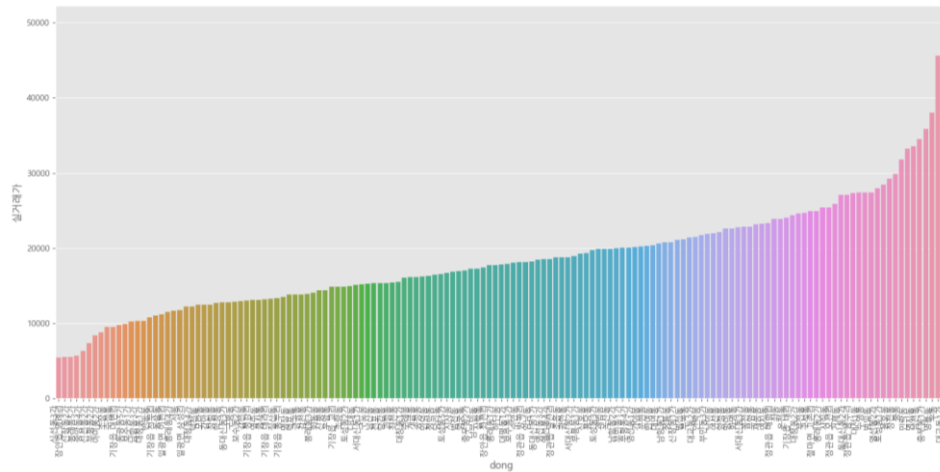
➤ 변수 'dong'

✓ 총 437개

상계동	29346
좌동	23255
화명동	21511
용호동	17398
중계동	17079

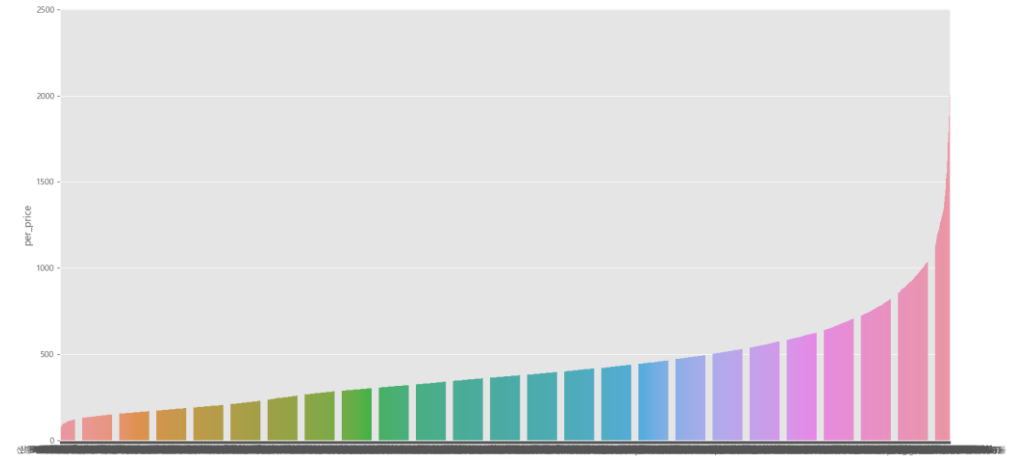
.....

누상동	2
주성동	2
구수동	1
옥인동	1
효제동	1



➤ 변수 'apt'

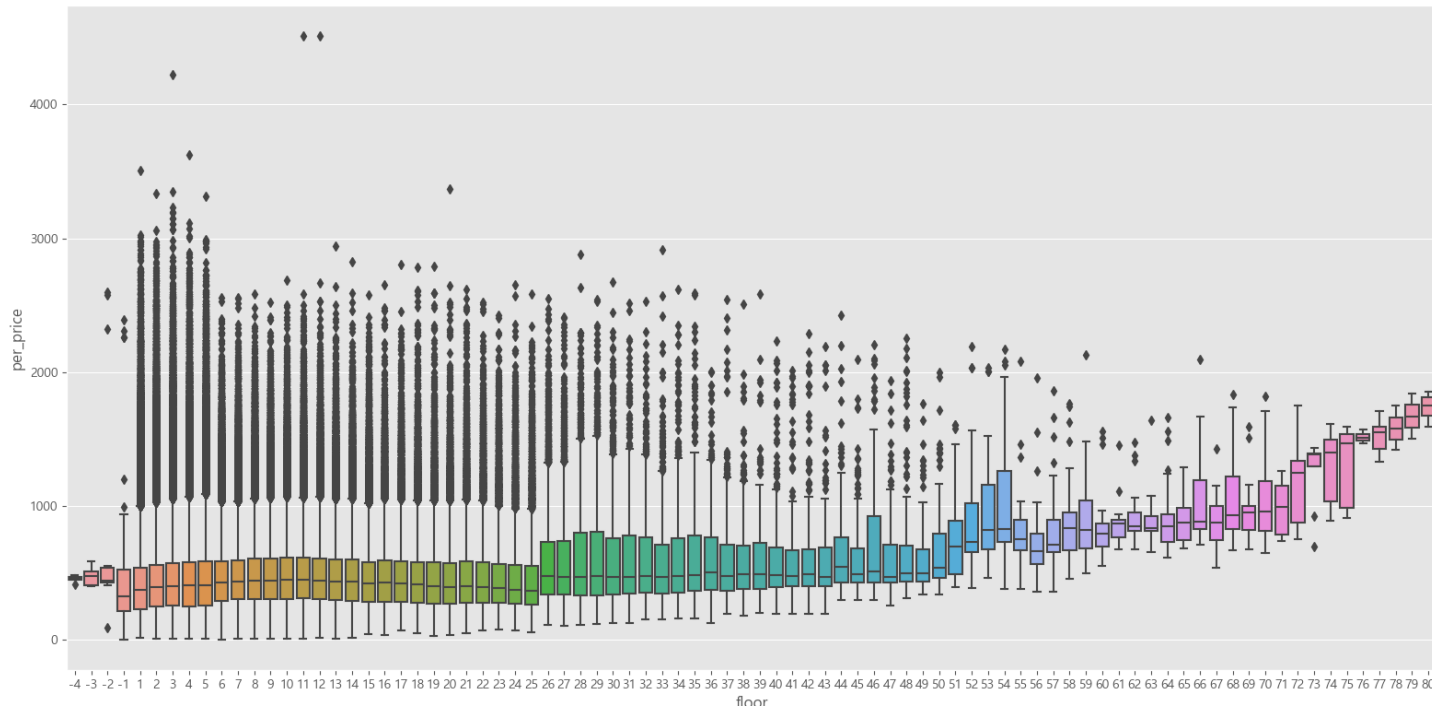
✓ 총 10,440개



✓ Labeling 과정 필요

EDA(Exploratory Data Analysis)

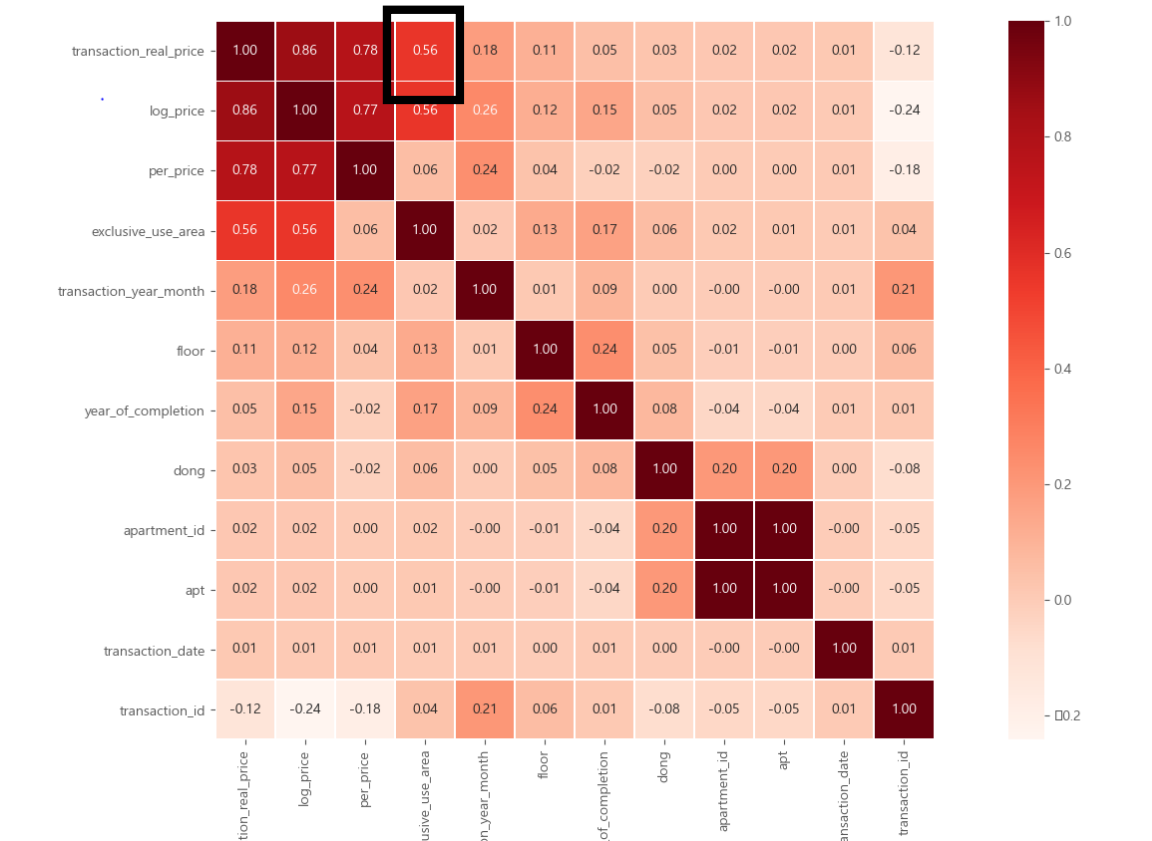
➤ 변수 'floor'



- ✓ 층이 올라갈수록 가격이 비싸지는 것을 가시적으로 확인
- ✓ 마이너스 층이라해서 가격이 크게 변동하지는 않음. 실제로 주택가의 지하는 가격이 낮게 형성되지만 아파트의 지하같은 경우는 평소 알던 지하의 개념이 아니라 실질적으로는 일반 층수와 똑같음

EDA(Exploratory Data Analysis)

➤ 변수 'exclusive_use_area'

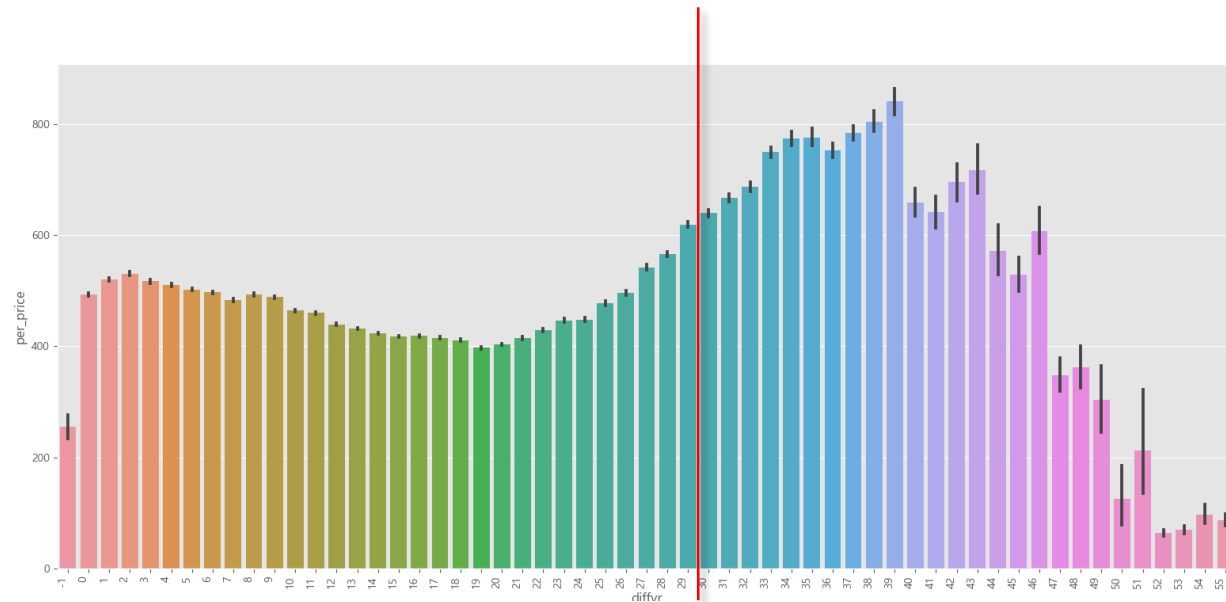


✓ 유일하게 실거래액에 대한 상관관계가 높게 형성 되어있음(면적이 넓을수록 가격이 올라감)

Preprocessing

➤ 파생 변수를 만들어 모델의 성능을 향상시킴

✓ 거래 년도 - 건립 년도(diffyr)



✓ 재건설 여부인 30년(old)

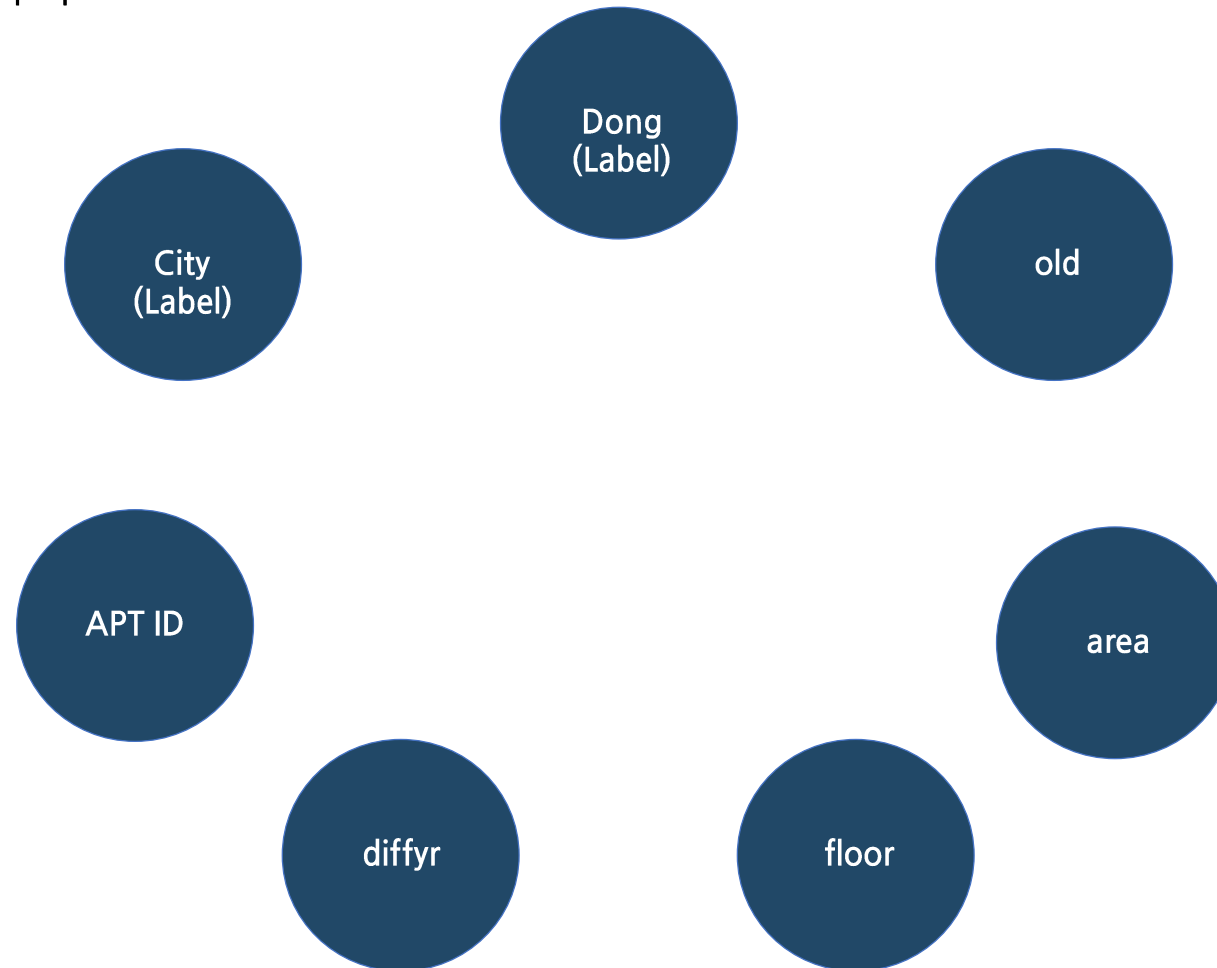
앞서 박근혜 정부는 2014년 '9·1 부동산 대책'으로 재건축 건축연한을 준공 후 40년에서 **30년으로 줄였다**. 규제가 완화되자 집값 상승으로 이어졌다. 시행 이후 3년간 강남·서초·송파 등 '강남3구' 재건축 **아파트값의 상승률이 약 30%에 달한다...**<뉴스핌 >

	df	Sum_sq	Mean_sq	F	Pr(>F)
Old	1	4.8e+09	4.8e+09	54975.43	0.0
Residual	1216551	1.0e+11	8.4e+04	NaN	NaN

✓ 두개의 파생 변수를 사용함으로써 성능이 더 좋은 것을 확인함

Feature Engineering

➤ 최적의 변수를 찾아서



Modeling

➤ 다양한 머신러닝 모델을 사용하여 최적의 예측값을 구함

규제 선형 모델

- Ridge : L2규제
- Lasso : L1규제
- Elastic Net : L1,L2 규제를 결합

회귀트리

- LightGBM

Model	CV score	RMSLE
Ridge	0.5255	0.4121
Lasso	0.5260	0.4121
ElasticNet	0.5260	0.4121
LightGBM	0.6517	0.3435

✓ DATA양이 많아 속도와 성능이 좋은 LightGBM을 최종 모델로 사용

Modeling


- GridSearchCV를 이용하여 하이퍼 파라미터를 찾아 최적의 예측값을 구함

LGBMRegressor 5 CV 시 최적 평균 RMSE 값 **0.3545**

최적 alpha: {'gpu_id': 0, 'max_bin': 80, 'max_depth': 128, 'n_estimators': 200, 'num_leave': 32, 'predictor': 'gpu_predictor', 'refit': True, 'tree_method': 'gpu_hist'}

최종 모델에 test data를 넣어 예측값을 구하고 다시 원값으로 변형 (np.exp())

Modeling

● WINNER ● 1% ● 4% ● 10%					
#	팀	팀 멤버	점수	제출수	등록일
24	yalthon		24538.04036	1	몇 초 전

✓ 아파트 가격에 영향을 미치는 지하철/학교/공원 등을 고려한 파생 변수를 만들어 놓으면 점수는 더 떨어질 것